МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 4**

по дисциплине: “Структуры и алгоритмы обработки данных”

на тему: ***”*** Алгоритмы поиска в линейных структурах ***”***

Вариант 5

Выполнил**:** студент группы *10701117* Мирзад Р.В.

Принял**:** Куприянов А.Б.

Минск 2019

**Цель работы**: изучить основные алгоритмы поиска в линейных структурах и научиться решать задачи поиска в линейных структурах на основе алгоритмов последовательного и *бинарного поиска*.

**Задания к лабораторной работе**.

Создать программу, выполняющую следующие действия.

1.Создание массива случайных чисел в диапазоне от -100 до 100, размер которого вводится с клавиатуры.

2. Сортировка созданного массива по возрастанию и подсчет количества операций перестановки.

3. Выполнение задания в соответствии с вариантом с помощью линейного и бинарного поиска и подсчет количества операций сравнения в каждом поиске.

4. Построить зависимости количества операций перестановки и сравнения от размера массива

При изменении размера массива от 10 до 100 с шагом 10.

Вариант 5

5. Найти и вывести на экран номер наименьшего элемента в массиве.

***Код программы***

***Сlass Program***

**import** random  
  
  
**class** MassBinAndLine():  
 **def** \_\_init\_\_(self, size):  
 self.a = []  
 self.size = size  
 **for** i **in** range(self.size):  
 self.a.append(random.randint(-100, 100))  
 print(self.a, **"неотсортированный массив"**)  
  
 *# Сортировка массива методом пузырька* **def** Sort(self):  
 countSwap = 1  
 **for** i **in** range(self.size):  
 **for** j **in** range(self.size - 1):  
 **if** (self.a[j] > self.a[j + 1]):  
 *# меняем элементы местами* temp = self.a[j]  
 self.a[j] = self.a[j + 1]  
 self.a[j + 1] = temp  
 countSwap += 1  
 print(self.a, **"Отсортированный массив"**)  
 print(countSwap,**" кол-во операций"**)  
  
  
 *# поиск номера минимального значения* **def** findingMinEl(self):  
 min = self.a[0]  
 index = 0  
 countSwap = 1  
 **for** i **in** range(self.size):  
 **if** self.a[i] < min:  
 min = self.a[i]  
 index = i  
 countSwap +=1  
 print(index, **'номер минимального значения\n'**,min,**'значение минимального элемента\n'**)  
 print(countSwap,**" кол-во операций"**)  
  
 *# линейный поиск* **def** lineFinding(self, key):  
 countSwap = 1  
 **for** i **in** range(self.size):  
 countSwap+=1  
 **if** self.a[i] == key:  
 print(**"номер введенного элемента"**, i, **"линейный поиск"**)  
 print(countSwap, **" кол-во операций"**)  
 **break  
 if** i == self.size - 1:  
 print(**"такого элемента не найдено линейным поиском"**)  
  
  
 *# бинарный поиск* **def** SearchBinary(self, el):  
 mid = len(self.a) // 2  
 low = 0  
 countSwap = 1  
 high = len(self.a) - 1  
 **while** self.a[mid] != el **and** low <= high:  
  
 **if** el > self.a[mid]:  
 low = mid + 1  
 countSwap += 1  
 **else**:  
 high = mid - 1  
 countSwap += 1  
 mid = (low + high) // 2  
 **if** low > high:  
 print(**"такого элемента не найдено бинарным поиском"**)  
  
 **else**:  
 print(**"номер введенного элемента"**, mid, **"бинарный поиск"**)  
 print(countSwap, **" кол-во операций"**)  
  
  
*# 5. Найти и вывести на экран номер наименьшего элемента в массиве.  
  
  
# бинарный поиск элемента***if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 m = MassBinAndLine(int(input(**"Введите размерность массива: "**)))  
 **while True**:  
 **try**:  
 print(**"--------------------------------------------\n"  
 "1.найти минимальный элемент\n"  
 "2.Сортировать массив\n"  
 "3.Поиск элемента линейным способом\n"  
 "4.Поиск элемента бинарным способом\n"  
 "5.выход"**)  
 user\_input = int(input(**"Выберите действие: \n"**))  
 **if** user\_input == 1:  
 m.findingMinEl()  
 **elif** user\_input == 2:  
 m.Sort()  
 **elif** user\_input == 3:  
 el = int(input(**"Введите искомый элемент: "**))  
 m.lineFinding(el)  
 **elif** user\_input == 4:  
 el = int(input(**"Введите искомый элемент: "**))  
 m.SearchBinary(el)  
 **elif** user\_input == 5:  
 **break  
 except** ValueError:  
 print(**"Введенное значение неправильно"**)

***Графики***

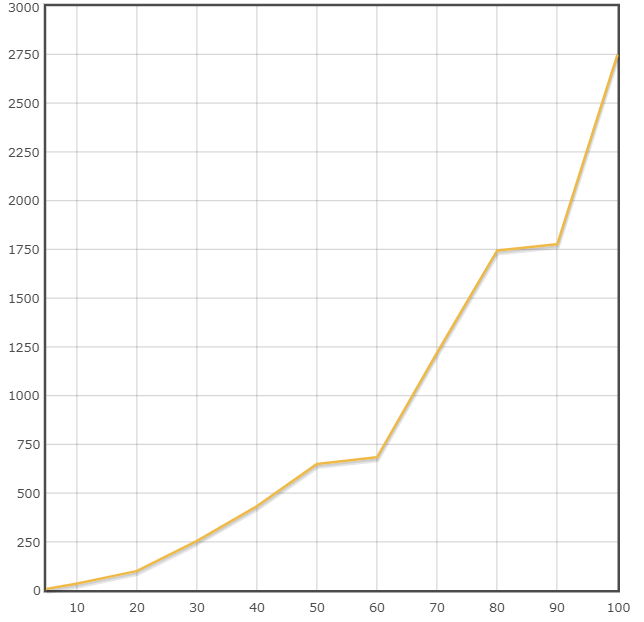
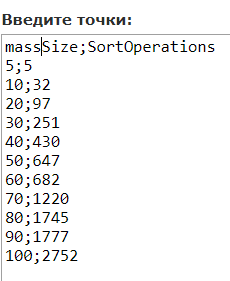


График зависимости операций перестановки от размера массива

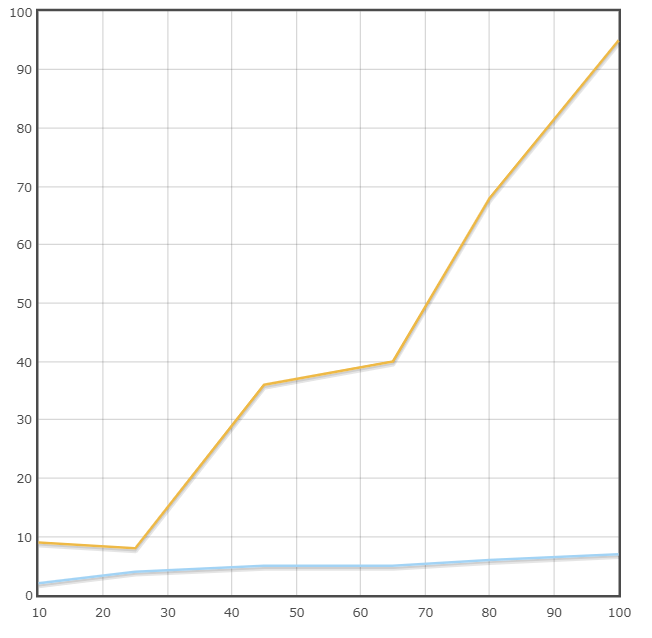
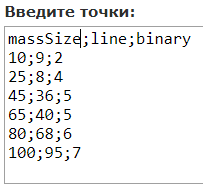
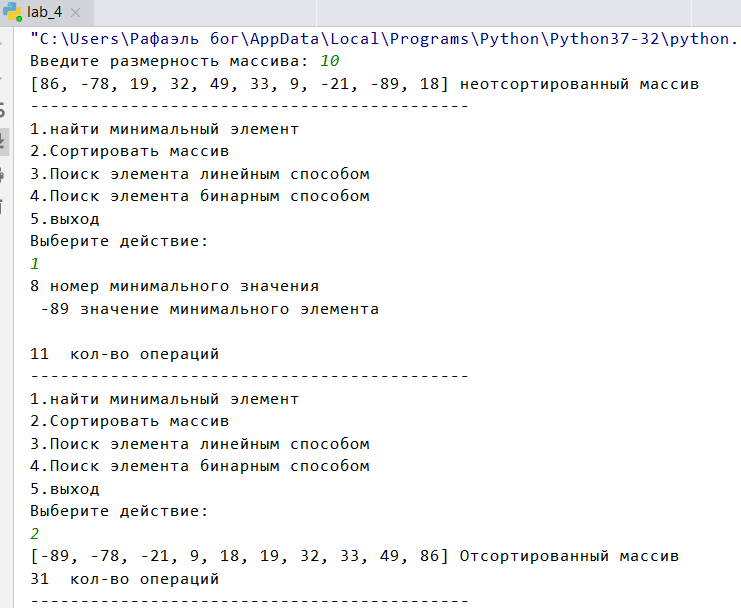


График зависимости поиска элемента линейным и бинарным способом

***Скриншот результатов***



**Вывод:** изучил основные алгоритмы поиска в линейных структурах и научился решать задачи поиска в линейных структурах на основе алгоритмов последовательного и *бинарного поиска*.